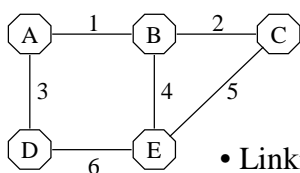


Linkkien tilaan perustuva reititys

Tavoitteena on välttää EV-protokollan luomat silmukat ja skaalautua laajempiin verkkoihin ja monenlaisiin topologioihin.

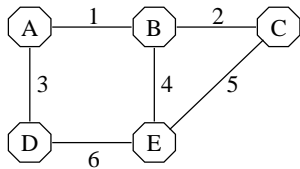
OSPF - Open Shortest Path First on Internetin linkkien tilaan perustuva reititysprotokolla



Esimerkkiverkko

- Linkintilaprotokolla ylläpitää verkon topologiakarttaa.
- Kun topologia muuttuu, kartat päivitetään nopeasti.
- Karttoja käytetään reitilaskennassa.
- OSPF on IETF:n määrittelemä linkintilaprotokolla Internetiä varten - OSPF on suositeltu RIP:n seuraaja.

Kartta esitetään täydellisenä linkkien luettelona



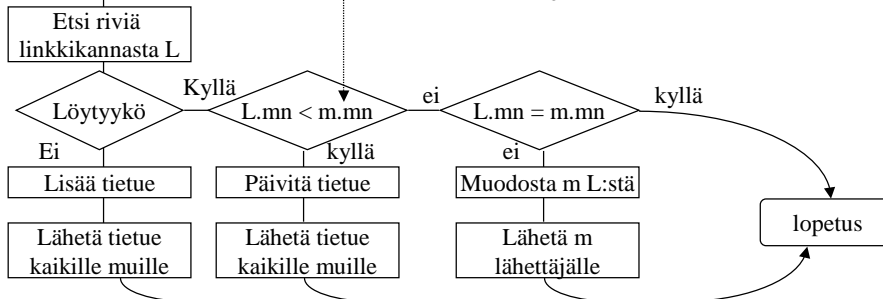
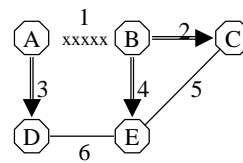
- Kustakin rivistä vastaa tietty solmu
- Linkin suunnat esitetään erillisinä

Mistä	Mihin	Linkki	Etäisyys
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Levitysprotokolla (flooding) levittää tiedot topologiamuutoksista

A:sta, B:hen, linkki 1, et = inf, muutosnumero

Levitysprotokolla on
 Vastaanota m
 kuittaa m

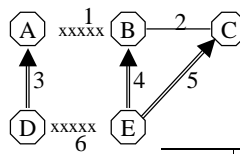


Linkkikanta AB vian levityksen jälkeen

Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

- Muutosnumeroiden laskenta alkaa käynnistyksessä 1:stä.
- Modulo aritmetiikka määrittelee mikä on "vähän suurempi kuin" --> muutosnumerolaskuri voi pyörähtää ympäri.

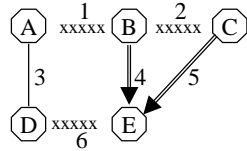
Kun verkko osittuu, puoliskojen kannat eroavat toisistaan



Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	inf	2
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

Mistä	Mihin	Linkki	Etäisyys	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Linkki 2 vikaantuu -> kannat eroavat lisää

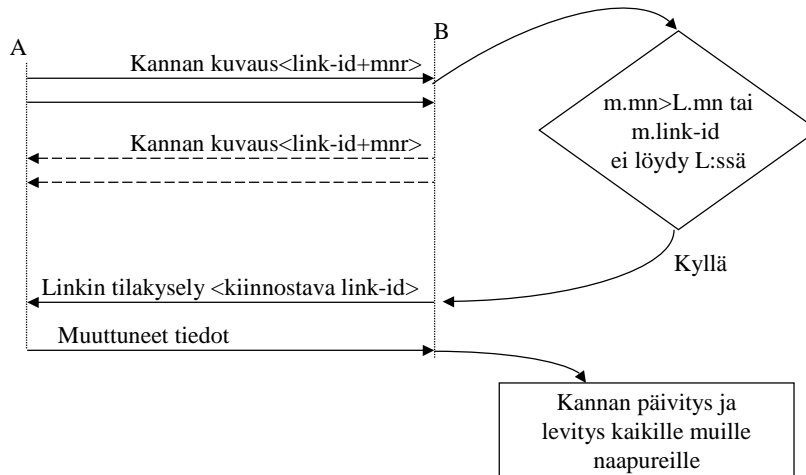


B:n, C:n ja E:n kannat:

Mistä	Mihin	Linkki	Etäisvvs	Mnumero
A	B	1	inf	2
A	D	3	1	1
B	A	1	inf	2
B	C	2	inf	2
B	E	4	1	1
C	B	2	inf	2
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	inf	2

Välittömästi tästä ei aiheudu ongelmaa, mutta jos linkki 1 elpyy ...

Osittuneen verkon jälleenyhdistyessä tarvitaan "naapurusten yhteenkasvatusta"



Jos reititin C uudelleenkäynnistyy

- Naapureilla voi olla C:n generoimia ja levittämiä ajalta ennen resettiä olevia tietueita
 - Resetin jälkeen C tarjoaa LSA muutosnumeroa InitialSequenceNumber,
 - Naapuri vastaa että sillä on uudempaa tietoa: $Mn > InitialSeqNr$
 - Mikäli C haluaa pitää kiinni LSA:staan, se kasvattaa $Mn = saatu-Mn + 1$ ja levittää LSA:nsa uudelleen
 - Mikäli naapaurin tieto ei vastaa resetin jälkeistä tilannetta, C vanhantaa tiedon asettamalla LSA:n iän arvoon Max-Age ja levittämällä tietoa uudestaan.

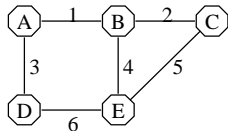
Linkkikantojen yhtenäisyys on varmistettava

- *Levityssanomien* kuitataan linkki kerrallaan
- Kannan *kuvaussanomien* perillemeno varmistetaan
- Kannan kukin tietue suojataan vanhenemisajastimella, jos päivitys ei saavu ajoissa, tietue poistetaan
- Kannan tietueet on suojattu tarkistussummalla
- Sanomat sisältävät autentikointitietoa
- *Mutta: päivityksen ollessa käynnissä toiset solmut ovat paremmin ajan tasalla kuin toiset --> reititysvirheitä*

OSPF perustuu Dijkstran SPF algoritmiin

- SPF - lyhin polku ensin -algoritmi laskee lyhimmän polun lähtösolmun S ja kaikkien muiden solmujen välillä
- Aluksi solmut jaetaan evaluoituihin E , joista alkavat polut tunnetaan ja muihin R .
- Lisäksi tarvitaan polkujen järjestetty lista O .

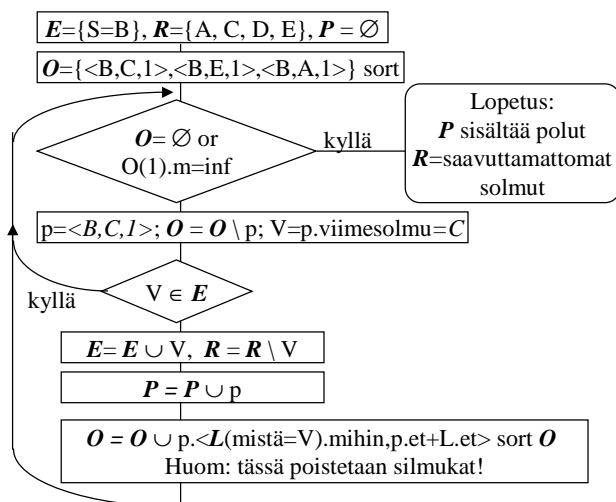
Dijkstran lyhin-polku-ensin algoritmi



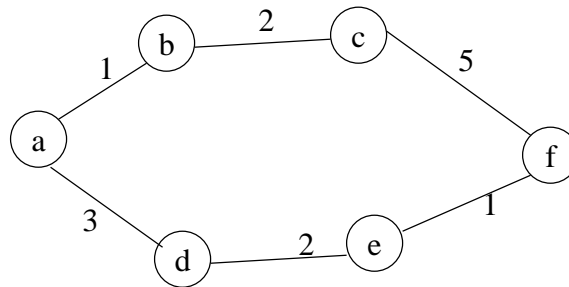
L

Mistä	Mihin	Linkki	Etäisyys
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Konvergoi nopeammin kuin Bellman-Ford
 $O(M \cdot \log M) < O(N \cdot M)$



Vaikka käytössä on erilaisia linkkipainoja SPF algoritmi löytää lyhimmät polut ensin



Lyhimmät polut a:sta f:ään: a-b; a-b-c; muodostetaan polku a-b-c-f, mutta se joutuu sortissa hännille; löydetään polku d:hen ja e:hen: a-d; a-d-e; muodostetaan polku a-d-e-f, joka ohittaa sortissa polun a-b-c-f. Lyhin polku f:ään on a-d-e-f!

Linkintilaprotokollien etuja ovat

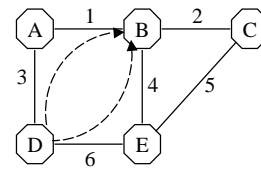
- Linkkikantojen tila konvergoituu nopeasti, ilman silmukoita
- Mittoina voidaan käyttää tarkkoja lukuja. Yksi protokolla tukee monta etäisyysmittaa
 - kapasiteetti, viive, hinta, luotettavuus.
- Ylläpitää useita polkuja yhteen kohteeseen.
- Ulkoisten polkujen erillinen esitys.

Useiden etäisyysmittojen käyttö edellyttää

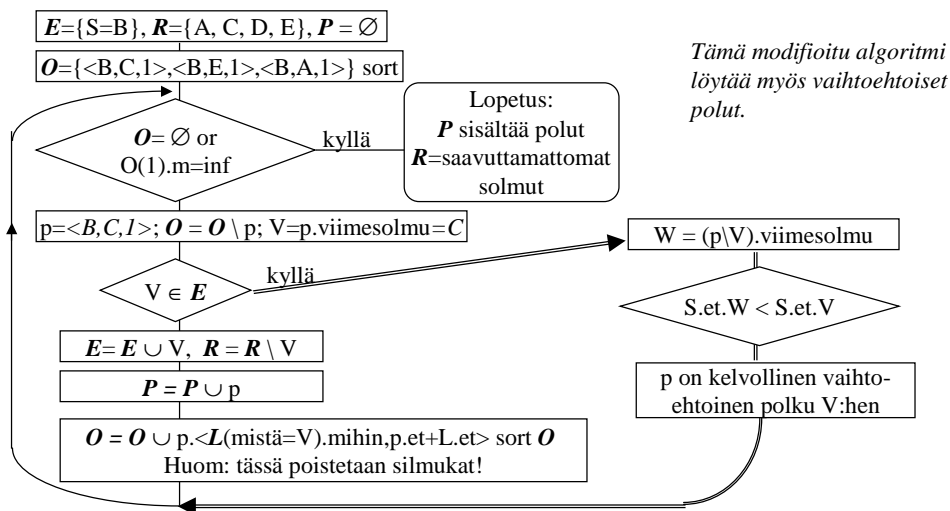
- Mittojen tallennusta jokaiselle linkille (*L.et1, L.et2 ...*)
- Erillisten reititystaulujen laskemista mitoille (*P(et1), P(et2) ...*)
- Linkkiprotokolla kuljettaa kaikki mitat
- Käyttäjäsanomiiin merkataan vaadittu mitta.
- Reittisilmukka on mahdollinen, jos solmut käyttävät eri mittaa samalle käyttäjän paketille.

Liikenteen jakaminen vaihtoehtoisille samanmittaisille poluille tehostaa verkon käyttöä

- Solmujen kokonaisjonopituus laskee
- Keskimääräinen viive laskee
- Päästä päähän viiveen vaihtelut pienenee
- Vikatilanteissa liikenteen heilahtelut vähenee
- Vaarana on pakettien järjestyksen muuttuminen, koska polkuviiheet (solmujen jonojen pituudet) vaihtelevat
- Hankaluus: menossa oleva liikenne ei pysy nykypolulla (no route pindown) --> epästabiilisuus mahdollista (2000)
- Milloin polut ovat riittävän samanmittaisia?



Sääntö $A \rightarrow Y \dots \rightarrow X$, jos $Y.et.X < A.et.X$ rajaa vaihtoehtoiset reitit (monotonisiin)

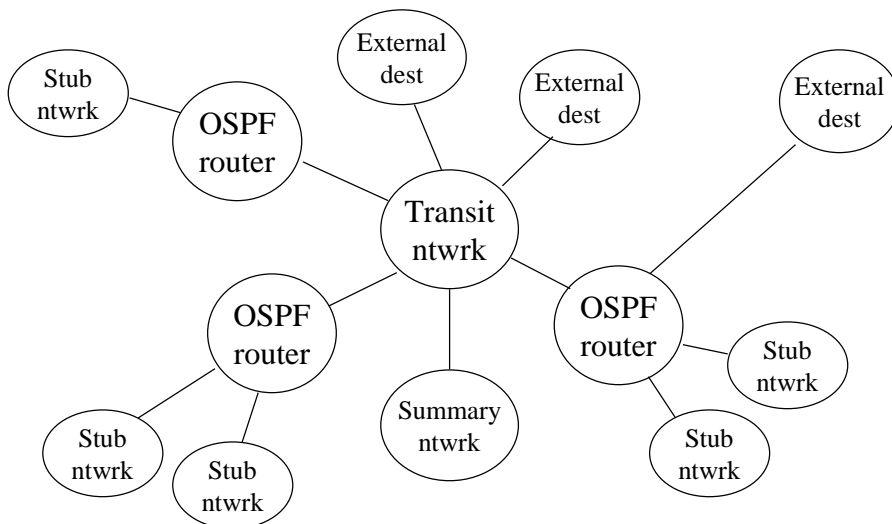


Linkintilaprotokolla kuvaa useita ulkoisia reittejä tarkoilla (halutuilla) mitoilla

- EV-protokollan mahdollisuudet kuvata reittejä ovat rajalliset äärettömyyteen laskennan ja Bellman-Fordin ($O(N^2)$) monimutkaisuuden takia
- Linkintilaprotokolla on vapaa yo rajoitteista. SPF reittilaskennan monimutkaisuus on $O(N \log N)$ - N =ulkoisten reittien määrä
- Esim. noin 30 000 ulkoista reittiä $\Rightarrow 10^9$ vs. 450 000

Tästä alkaa OSPF protokollan periaatteiden kuvaus

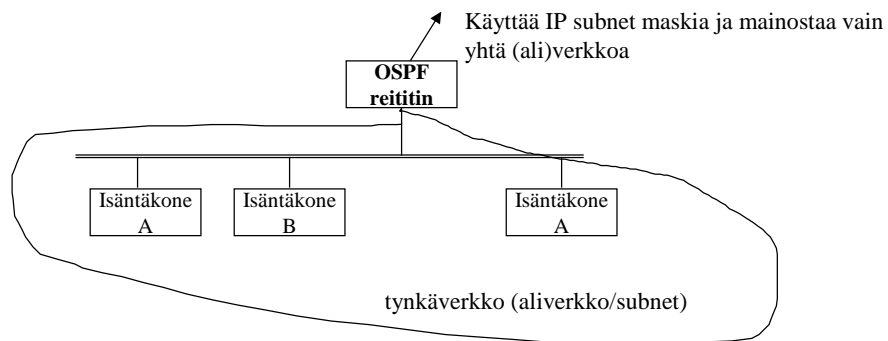
OSPF sees the network as a graph



OSPF:ssä on kolme osaprotokollaa

- Hello protokollan (huomio protokolla) avulla muodostetaan naapuruussuhteet ja valvotaan yhteyksiä naapureihin
- Tiedonvaihtoprotokollan (Exchange protocol) avulla tapahtuu naapurusten yhteenkasvattaminen eli tietokantojen synkronointi
- Levitys (flooding) protokollan avulla tilamuutokset levitetään kaikille solmuille.

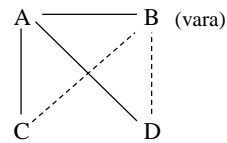
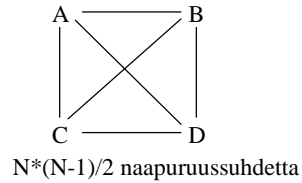
OSPF erottelee reitittimet ja isäntäkoneet



OSPF tukee yleislähetysverkkoja

Yleislähetysverkossa

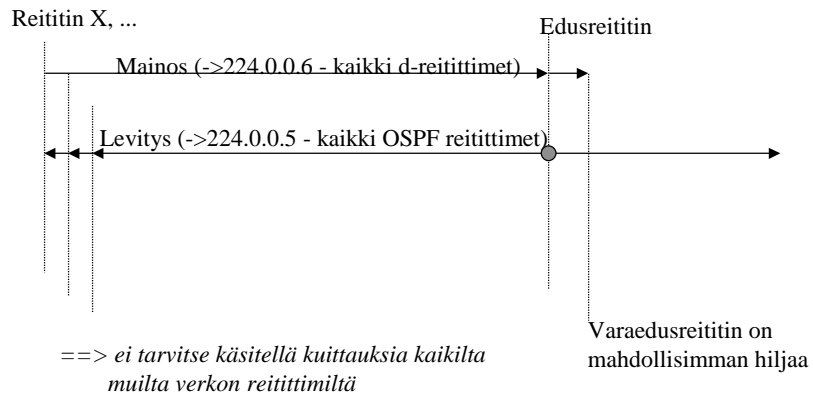
- kaikki voivat lähettää kaikille
- yksi voi lähettää kaikille tai joukolle
- jos siinä on N reitintä, niillä on $N*(N-1)/2$ naapuruussuhdetta ja
- jokainen reititin mainostaa $N-1$ reittiä toisiin reitittimiin + yhtä tynkäverkkoa



Naapuruus rajattu vain edus- (designated) reitittimeen (A) ==>

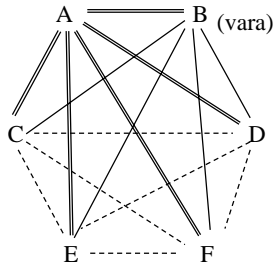
- A täytyy valita Hello-protokollan avulla
- Linkkikantojen synkronointi yksinkertaistuu
- Varaedus (backup designated) reititin valitaan samalla kertaa

OSPF levitysprotokolla yleislähetysverkossa

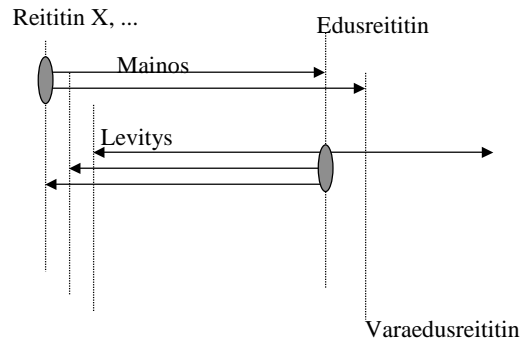


Ei-BC verkoissa OSPF toimii samoin kuin BC-verkoissa paitsi, että BC-lähetykset korvataan yksipistelähetyksillä

Edusreititin

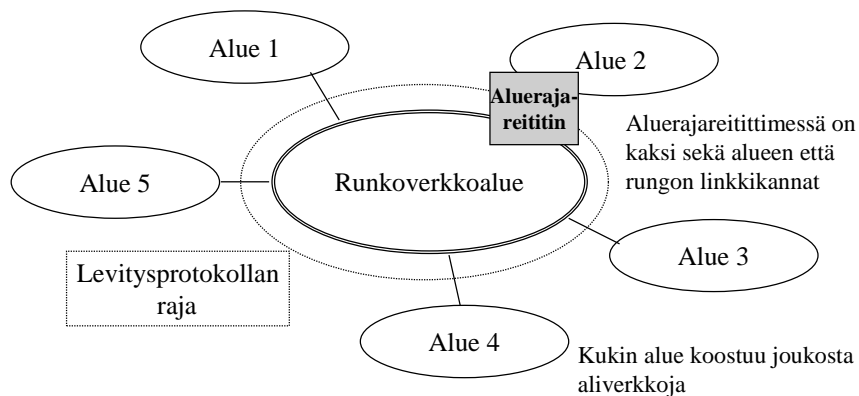


- Kiinteä yhteys edusreitittimeen
- Kiinteä yhteys varareitittimeen
- Valinnainen yhteys muiden reitittimien välillä



Huom: kiinteitä yhteyksiä kannattaa minimoida, koska ne ovat kalliita

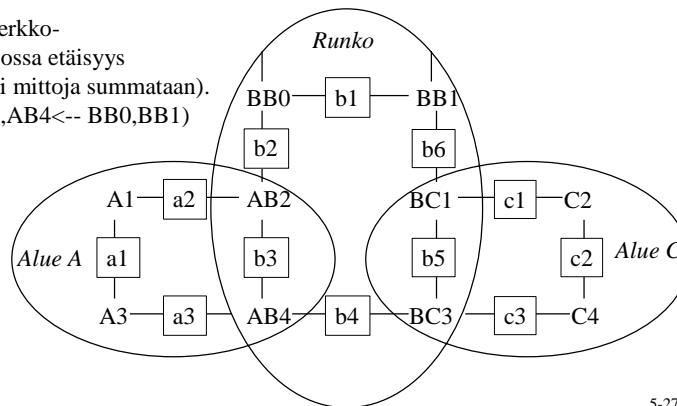
Jakamalla laaja verkko alueisiin OSPF helpottaa levitystä ja pienentää linkkikantoja



Muiden alueiden (ali)verkot kuvataan yhteenvetotietueissa, joiden mitta lasketaan RIP:n tapaan

Alueen A linkkikanta:

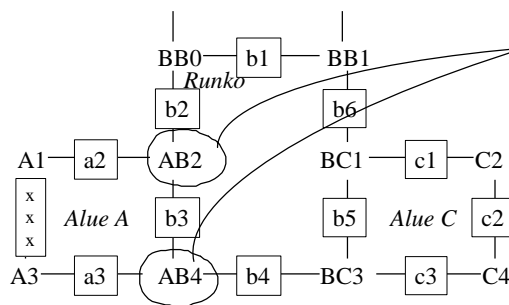
- a1
- a2
- a3
- rungon ja Alueen C aliverkkotietueet (<- AB2,AB4), jossa etäisyys ABx--bz tai ABx--cy (eli mittoja summataan).
- ulkoiset tietueet (<-AB2,AB4<-- BB0,BB1)



S38.121/RKa s-01

5-27

OSPF elpyy helposti alueiden vikatilanteista



AB2 ja AB4 mainostavat vain niitä verkkoja, jotka ne todella voivat saavuttaa: AB2: a2 ja AB4: a3. Runko ei tunne Alueen A tarkkaa rakennetta, vaikkakin se tuntee A:n kaikki saavutettavat aliverkot.

S38.121/RKa s-01

5-28

LSA types in OSPF are

- LS Type = 1 Router LSA -- describes set of active If and neighbors
- LS Type = 2 Network LSA -- describes a network segment (BC or NBMA) along with the IDs of currently attached routers
- LS Type = 3 Summary LSA --
- LS Type = 4 AS Border Router summary LSA
- LS Type = 5 AS- external LSA -- descr ext routes
- LS Type = 6 Group Membership LSA (MOSPF - Multicast)
- LS Type = 7 NSSA LSA -- to import limited external info
- LS Type = 8 (proposed) external attributes LSA (in lieu of Internal BGP)

Hierarchical
Routing

Alueen OSPF reitittimillä on samansisältöinen linkkien tilatietokanta

Yhteinen linkintilamainoksen otsikko

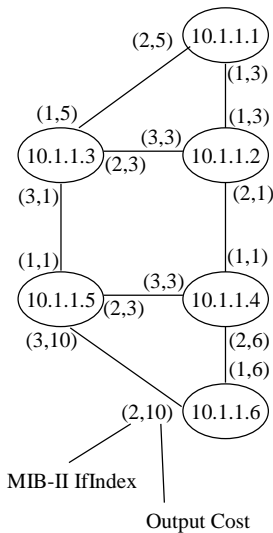
- Kannassa on 7 "LS" tietuetyyppejä
1. reititin (router LSA)
 2. Verkko (network LSA)
 3. IP-verkon yhteenveto
 4. reunareitittimen yhteenveto
 5. ulkoinen
 - .. yhteenvedoilla on sama formaatti
 6. Monilähetys LSA
 7. NSSA tietue
 - jne ...

LS ikä	optiot	LS tyyppi
Linkin tila ID		
Mainostava reititin		
LS järjestysnumero		
LS tarkistussumma	pituus	

Avain

LS ikä - sekunteina mainoksesta
 optiot: E - ulkoiset linkit
 T - palvelun tyyppi -- eri mittoihin
 perustuva
 reititys

Example of the Router LSA



0 seconds
0x9b47
0 (ordinary)
1 (pt-t-pt), 0
1 (pt-t-pt), 0
3 (stub netw),
0 Tos metrics

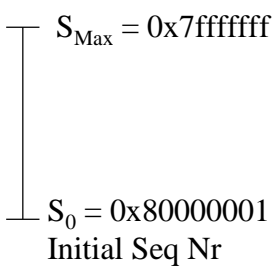
LS Age	Options	LS Type
Link State ID		
Advertising Router		
LS Sequence Number		
LS Checksum	Length	
Router Type	0	Nrof links
Link ID		
Link Data		
Lnk Type	#Tos met	Metric
Link ID		
Link Data		
Lnk Type	#Tos met	Metric
Link ID		
Link Data		
Lnk Type	#Tos met	Metric

E-bit, LS Type 1,
(Router LSA)
10.1.1.1
10.1.1.1
0x80000006
60 bytes
3
10.1.1.2 (Neighb)
IfIndex 1 (Unnum)
3
10.1.1.3 (Neighb)
IfIndex 2 (Unnum)
5
10.1.1.1
255.255.255.255
0

Router 10.1.1.1's router-LSA

Length = 24 + 3 * 12 = 60 bytes
Router with 100 interfaces:
length = 24 + 100 * 12 = 1224 bytes

LSA Sequence Numbers

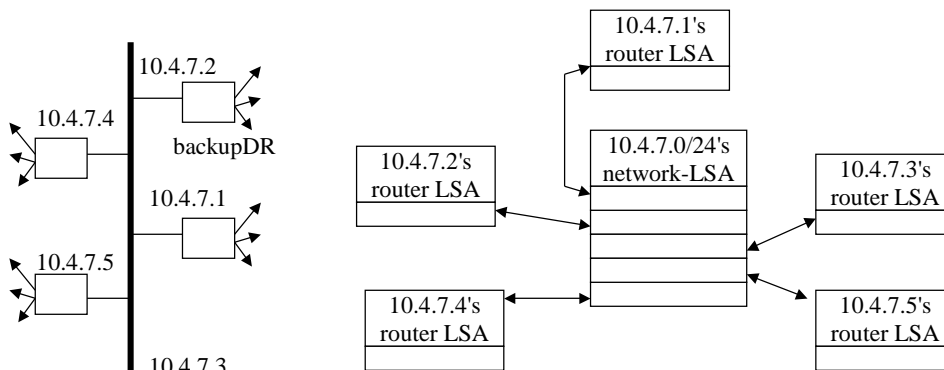


- To roll the space over, first delete record with S_{Max}
- A router may update a self originated record only once in 5 sec.
- In absense of errors rolling the space over takes at least 600 years.
- LS Age is updated during flooding at each step. Records with max Age are discarded. This breaks inf loops.

OSPF timeouts - LS Age field

Constant	Value	Action of OSPF router
MinLSArrival	1 second	Max rate at which a router will accept updates of any LSA via flooding
MinLSInterval	5 seconds	Max rate at which a router can update an LSA
CheckAge	5 min	Rate to verify an LSA Checksum in DB
MaxAgeDiff	15 min	When Ages differ more than 15 min, they are considered separate. Smaller LS age - newer!
LSRefreshTime	30 min	A Router must refresh any self-originated LSA whose age has reached 30 min.
MaxAge	1 hour	LSA is removed from DB.

Network-LSA reduces LinkDB for BC networks



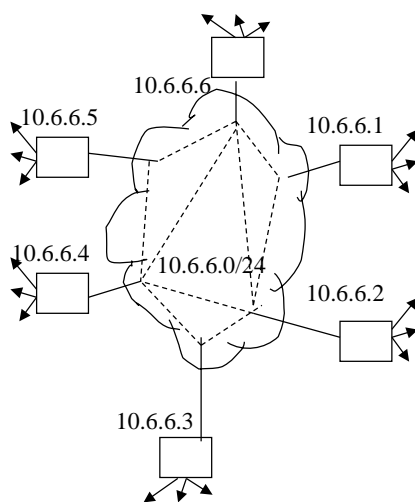
Network-LSA is generated by DR
 Network-LSA reflects DB sync status with DR both ways
 This reduces nrof link records from $O(n*(n-1))$ to $n*2$.
 Particularly important if Network is ATM or FR with a lot of routers attached!

DR - designated router

Nonbroadcast Multiaccess(NBMA) subnets support many routers communicating directly but do not have BC capability

- Examples are ATM, Frame Relay, X25
- IP routing requires more manual configuration
- Designated router and backup DR concept reduce the number of adjacencies
- The model is prone to failures that may be hard to track

Point-to-multipoint subnet is more robust but less efficient



- There is no DR nor backup DR
- Every OSPF router maintains adjacencies with all neighbors with whom it has direct connectivity
- Alternative is a set of NBMA networks
- Next hop routing protocol improves scalability

OSPF packets - the protocol itself

- OSPF works directly on top of IP. OSPF protocol number is 89.
- For most packets TTL = 1, except for hierarchical routing
- Dest IP address = Neighbors IP address or AllSPFRouters (224.0.0.5) or AllDRouters (224.0.0.6)
- Packet types are
 - Type 1: Hello
 - Type 2: Database Description packet
 - Type 3: Link State Request packet
 - Type 4: Link State Update packet
 - Type 5: Link State Acknowledgement packet

OSPF protokolla toimii suoraan IP:n päällä

OSPF:ssä on 3 osaprotokollaa:

- Huomio (Hello) protokolla
- Tiedonvaihto (exchange) protokolla
- Levitys (flooding) protokolla

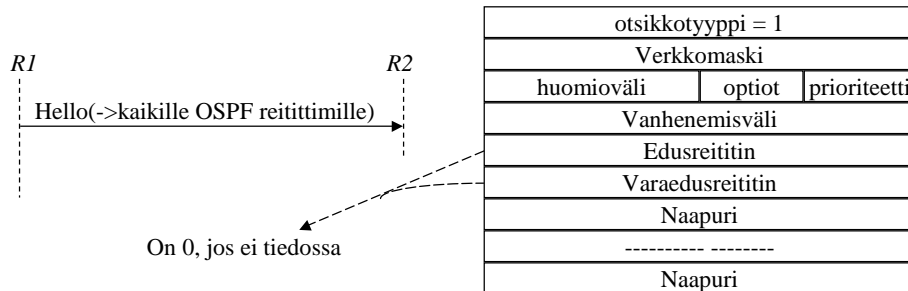
OSPF sanomien yhteinen otsikko on:

Versio	Tyyppi	paketin pituus
Reitittimen ID		
Alueen tunnus		
Tarkistussumma	Autentikointityyppi	
Autentikointi		
Autentikointi		

OSPF nykyversio on 2.

Tyyppi erottelee OSPF sanomat toisistaan. Autentikointi perustuu salasanoihin, jolloin sen merkitys on rajallinen tai uudempiin kryptografisiin menetelmiin (1995).

Huomioprotokolla varmistaa, että linkit toimivat ja valitsee edus- ja varaedusreitittimen



- Naapuri - lista naapureista, joilta on tullut viestejä viimeisen vanhenemisvälin aikana
- Huomioväli kertoo, kuinka usein paketteja lähetetään.
- Optiot - E -ulkoisia linkejä, T - TOS reititys onnistuu.
- Prioriteetti kertoo kelpoisuudesta edusreitittimeksi.
- Huomioviestien pitää kulkea linkillä molempiin suuntiin, jotta linkki kelpaisi reitiksi

S38.121/RKa s-01

5-41

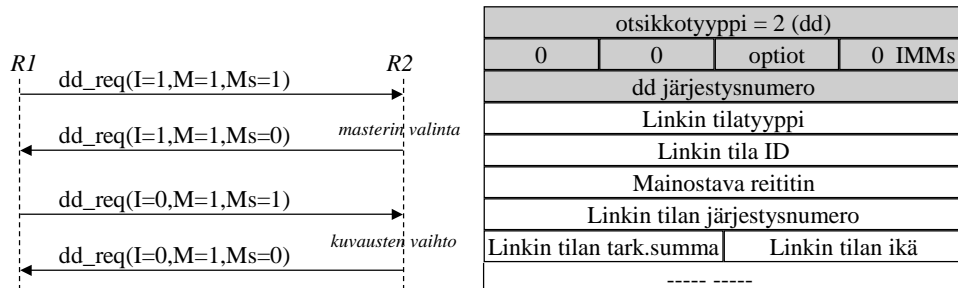
Huomioprotokollan avulla valitaan edus- ja varaedusreititin

1. Vaalikelpoisuus saavutetaan yhden vanhenemisvälin jälkeen mikäli kaksisuuntainen saavutettavuus on OK.
2. Varaedisreitittimeksi valitaan korkeimman prioriteetin omaava niistä, jotka ilmoittautuivat. Tasatilanteessa valitaan suurimman ID:n omaava.
3. Jos kukaan ei ilmoittautunut varaedisreitittimeksi, valitaan korkeimman prioriteetin omaava naapuri. Tasatilanteessa suurimman ID:n omaava.
4. Edusreitittimeksi valitaan ilmoittautuneista yo. säännöllä.
5. Jos yksikään ei ilmoittaudu edusreitittimeksi, varaedisreititin ylennetään. Kohdat 2 ja 3 suoritetaan uudestaan varaedisreitittimen valitsemiseksi.
6. Muutoksia minimoidaan siten, että korjattu entinen edusreititin ei kiirehdi ilmoittautumaan uudelleen. Kohtia 2....5 suoritetaan jatkuvasti.

S38.121/RKa s-01

5-42

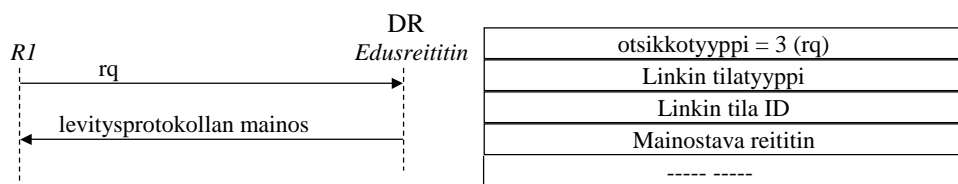
Vaihtoprotokolla alkusynkronoi linkkikannan edusreitittimen kannan kanssa



- Master lähettää oman kantansa kuvauksen järjestysnumeroiduissa paketeissa
- Slave kuittaa paketit lähettämällä vastaavat omat kuvauksensa.
- Vaihto jatkuu kunnes molemmat ovat lähettäneet omat kuvauksensa.
- Erot kirjataan kyseltävien tietueiden listaan.

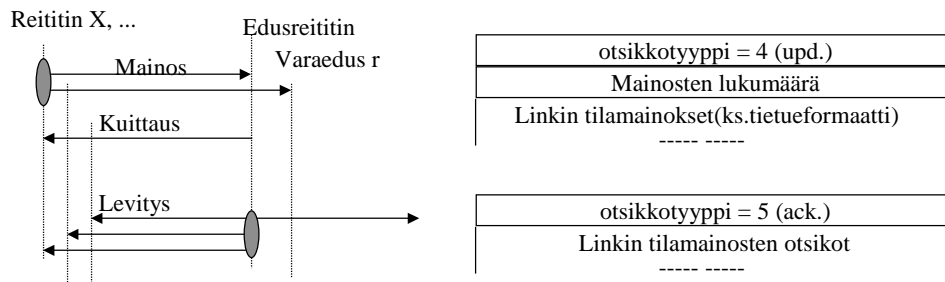
I = initialize
M= more
Ms = Master/slave
(aloituspaketti on harmaa)

Tietueiden sisältö kysellään pyyntöpaketeilla, jotka kuitataan levitysprotokollan paketein



- Vastausta odotetaan uudelleen lähetyksen väli, jos ei vastausta, pyyntö toistetaan.
- Jos kyseltävät tietueet eivät mahdu yhteen pakettiin, tietueet jaetaan useaan kyselyyn.
- Jonkin mennessä pieleen, palataan herkästi roolien uudelleen neuvotteluun.
- Tietuesisältöjen kysely voi alkaa heti, kun yksikin eroava tietue on havaittu, jolloin dd-pakettien ja rq pakettien vaihto tapahtuu rinnan.

Levitysprotokolla ylläpitää alueen linkkikantojen yhtenäisyyttä



- Alkuperäisen mainoksen lähettää aina linkistä vastaava reititin.
- Mainosta levitetään levityssääntöjen mukaan koko alueelle ($\text{age} = \text{age} + 1$).
- Uuden tietueen kuittaus voidaan BC verkossa korvata levityssanomalla
- Yksi ack voi kuitata monta mainosta.

Linkkitietueilla on ikä, vanhat poistetaan kannasta

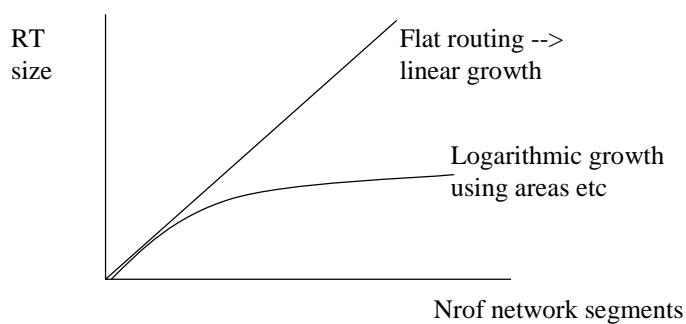
1. Ikä = mainosten kulkemien linkkien lkm + sekunnit vastaanotosta
2. Maksimi-ikä = 1 tunti
3. Jokaista tietuetta pitää mainostaa vähintään 30 min välein.
Uusi mainos nollaa iän ja inkrementoi tietueen järjestysnumeron.
4. Kun ikä tulee täyteen ($\text{MaxAge}=1\text{h}$), lähetetään mainos.
5. MaxAge mainos hyväksytään ja levitetään - tämä poistaa vanhan tiedon.
6. Jos mainoksen ikäero kantaan on pieni mainosta ei levitetä, jotta saman tiedon kopiot eivät kuormittaisi verkkoa
7. Jos MaxAge tietuetta ei löydy, mainos ei aiheuta toimenpiteitä, (koska reititin on jo ehtinyt poistaa vanhan tiedon.)

Summary of OSPF subprotocols

	Hello msg (1)	DD (2)	LS rq (3)	LS upd (4)	LS ack (5)
Hello protocol	X				
Database exchange		X	X	X	X
Flooding protocol				X	X

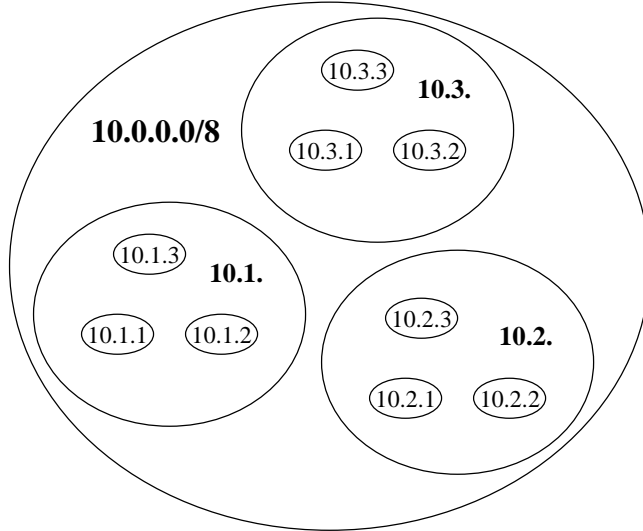
OSPF without Dijkstra's algorithm and with more generic data objects is SCSP (Server Cache Synchronization Protocol) which is studied as the basis for *Telephony Routing Information Protocol* - in our Lab.

The purpose of hierarchical routing in OSPF is to reduce routing table growth



The cost is: sometimes sub-optimal routes.

Example of use of routing hierarchy

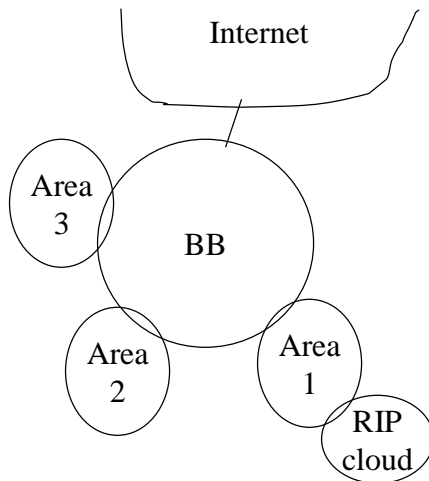


- Example:
- 16 segments in each lowest level network
 - flat routing:
RTsize= $16 * 9 = 144$
 - areas 10.1.1:
16 local routes +
10.1.2/24
10.1.3/24
10.2/16
10.3/16
== 20 RT entries!

S38.121/RKa s-01

5-49

OSPF supports 4 level routing hierarchy



Level	Description
1	Intra-area routing
2	Inter-area routing
3	External Type 1 metrics
4	External Type 2 metrics

- Type 1 metrics are of the same order as OSPF metrics, e.g. hop count (for RIP and OSPF)
- Type 2 metrics are always more significant than OSPF internal metrics

S38.121/RKa s-01

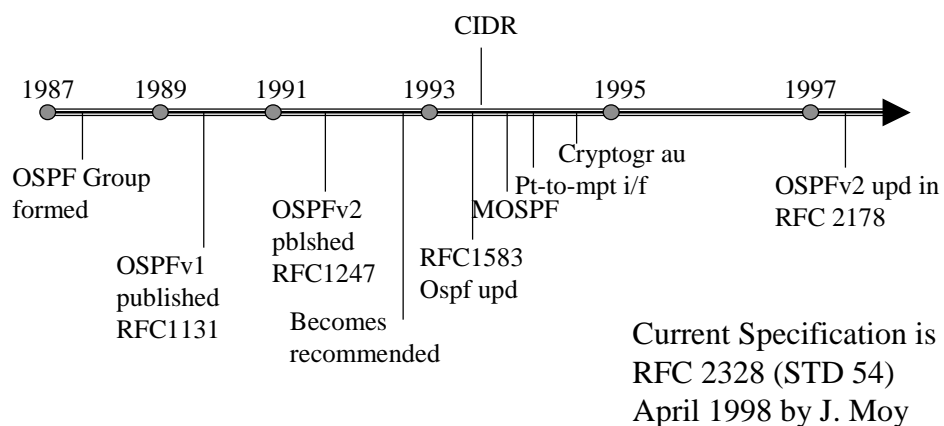
5-50

Why is it difficult to route packets around network congestion?

- BBN ARPANET link state metric varied with the length of the output queue of the link --> lead to route trashing.
- The problem is there is no route pin-down for existing traffic.
- By limiting the range of the metric changes, an equilibrium could be reached. Nevertheless routing instability is the problem.

When QoS or Class of Service a'la DiffServ is introduced this problem again becomes important.

OSPF development history



CIDR - Classless Inter Domain Routing

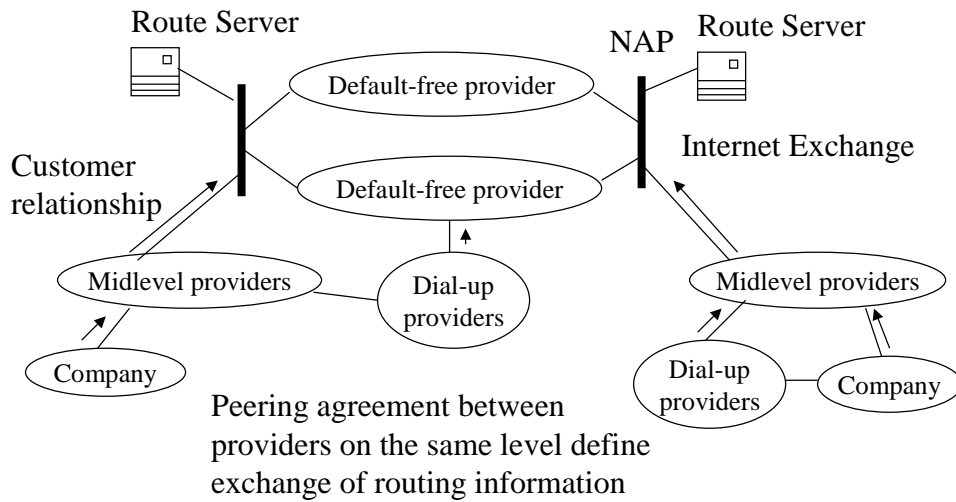
Internetin kasvu on pakottanut ottamaan käyttöön CIDR osoitearitmetiikan, jolla IP-osoitteavaruus saadaan tehokkaampaan käyttöön.

Forwarding process
Router implementation

CIDR vaikuttaa useisiin reititysprotokolliin

- AS - Autonomous System on Internetin hallinnollinen alue, eli osa verkosta, jolla on yksi omistaja.
- AS:lla käytössä on yleensä yksi (sisäinen) reititysprotokolla, esim OSPF.
- AS välillä käytetään ulkoista reititystä.

Organisation of the Internet as Autonomous Systems



S38.121/RKa s-01

5-55

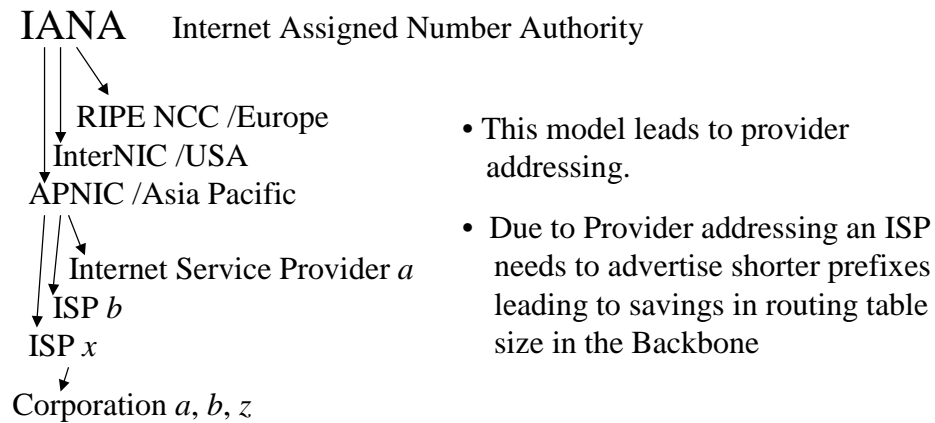
History of the Internet Core

-1985 Arpanet
-1987 NSFNET 56k lines
-1992 NSFNET T1 lines (1.5M)
- 1995 NSFNET T3 lines (24M)
- 1995 NSFNET decommissioned
- 1995... Commercial (UUNET, MCI, Sprint...)

S38.121/RKa s-01

5-56

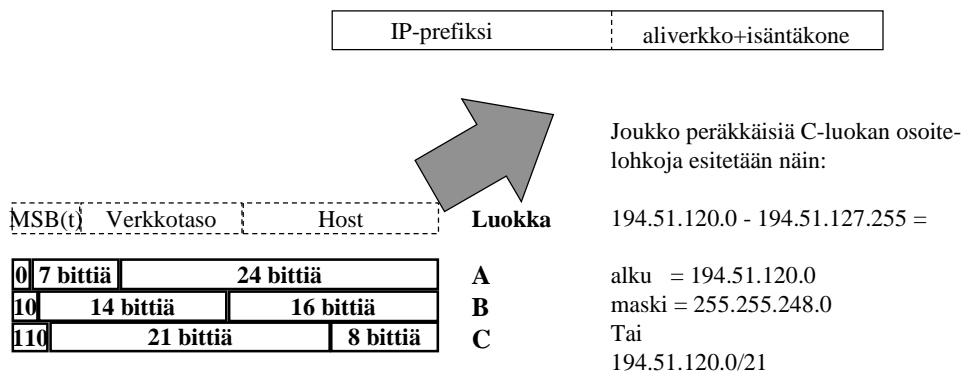
Internet Addresses are assigned by a hierarchy of registrars



S38.121/RKa s-01

5-57

CIDR aritmetiikka perustuu 32-bitin IP-osoitteen jakamiseen vapaasti verkko- ja häntäosaan



S38.121/RKa s-01

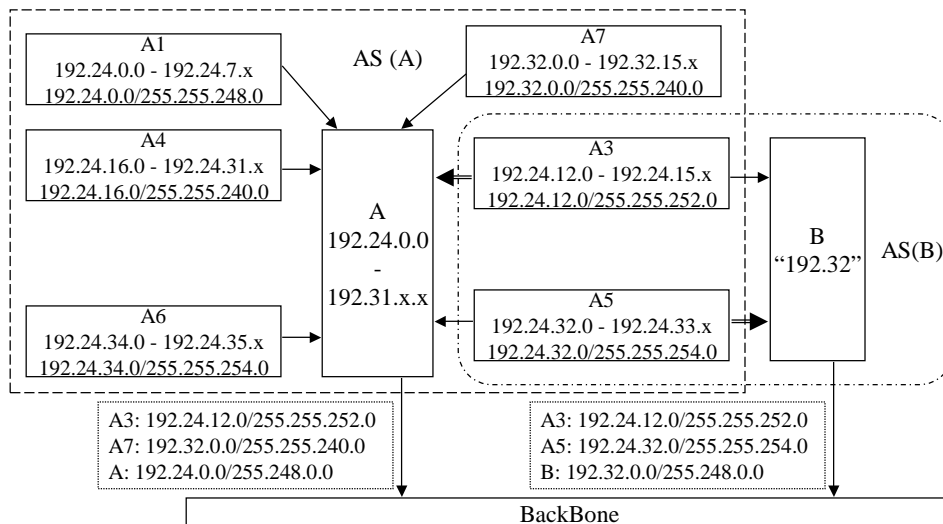
5-58

CIDR muuttaa reittien mainostusperiaatetta

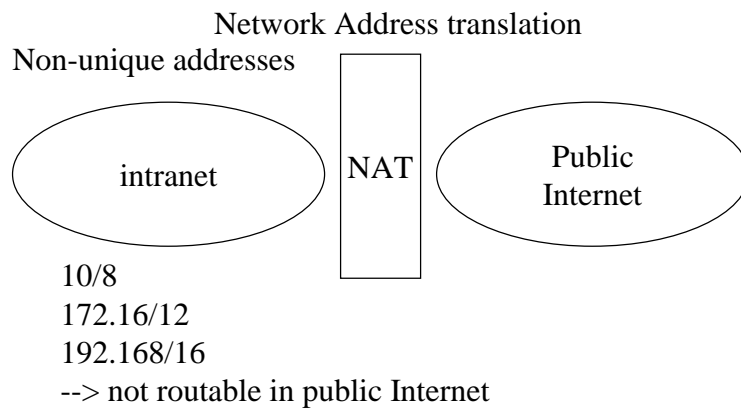
Sääntö 1: Reititys kohteeseen tapahtuu aina pisimmän matchaavan osoitteen perusteella. ---> useaan AS:ään liittyvien (moni-kotisten) verkkojen osoitteita ei voi aggregoida tavalliseen tapaan.

Sääntö 2: Verkko, joka aggregoi useita reittejä, tuhoaa paketit, jotka matchaavat summattuun osoitteeseen, mutta eivät yhteenkään summan tekijään. (tällä estetään silmukoiden muodostumista).

Esimerkki



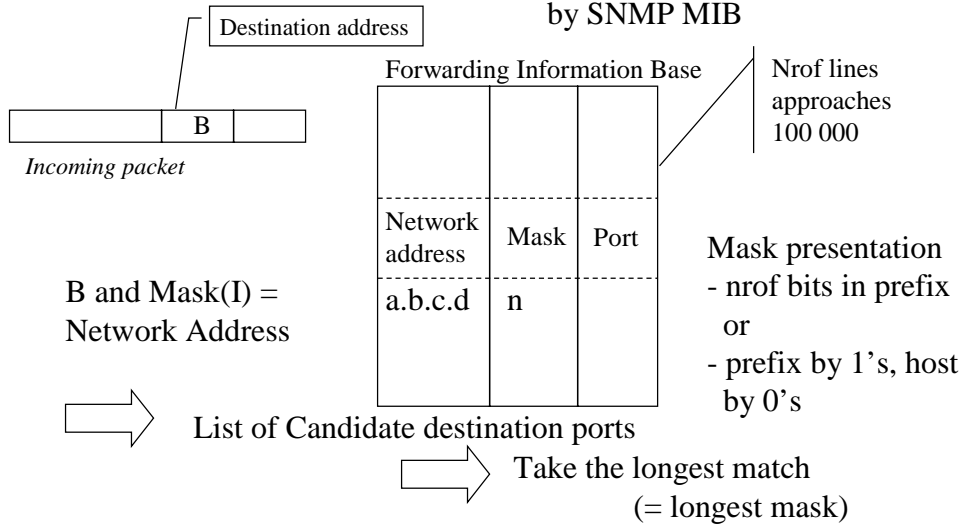
Network Address Translation (NAT) preserves address space and improves security



Packet Forwarding and Router Architectures

Packet forwarding in a router

RFC 2097 - view routing table
by SNMP MIB



S38.121/RKa s-01

5-63

Modifications of forwarding process

- Multipath routing
 - e.g. hash(source IP address, destination IP address) produces one of the possible next hops.
- TOS routing
 - never widely used
 - has been removed from recommendations (RFC 2178 in 7/97)
- Source Routing (strict or loose)
 - max 9 hops can be specified in header options
 - has performance penalty
 - is considered a security hole (all packet may be dropped)

S38.121/RKa s-01

5-64

More modifications of the forwarding process

- When there are too many packets to forward, some need to be dropped. To maintain a fair service drop algorithms are used
 - e.g Random Early Detection (RED)
- Scheduling algorithms manage the share of connections in the available bandwidth
 - e.g give 15 kbit/s to an audio conference or half of the link bandwidth to interactive services
 - Weighted Fair Queuing (WFQ) and Class Based Queuing (CBQ) are examples of scheduling algorithms

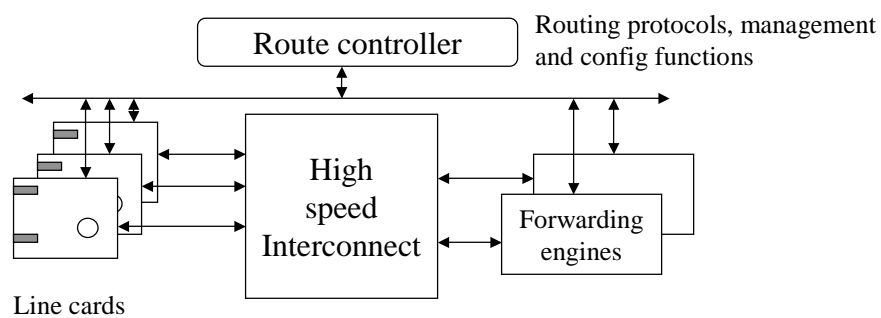
Routers support Security and problem resolution

- Security includes e.g. preventing unauthorised access to a company intranet
 - we talk about Firewalls
 - forwarding needs to check filtering rules on IP addresses and TCP port numbers
 - ISP routers may check all source IP addresses to trace security attacks
- A router may support RMON MIB
 - router allows traffic tracing for routing problem analysis

Routers can collect Statistics

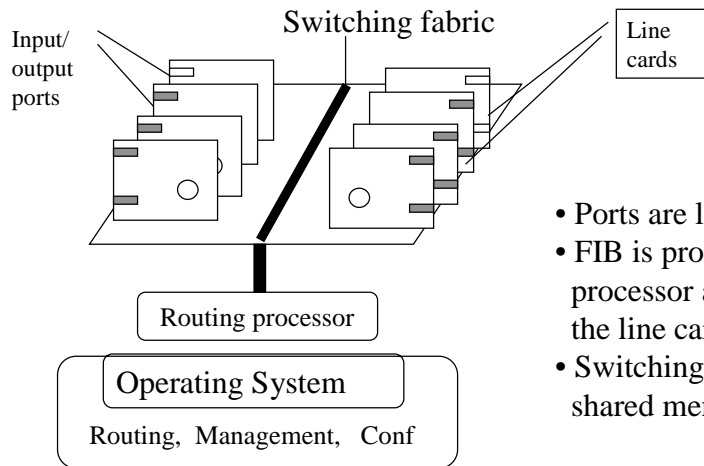
- Statistics are needed for Network Planning,
- Inter ISP accounting and
- Usage based charging

A non-trivial router architecture is



Forwarding capacity is increased by sharing the load between several forwarding engines

A faster router architecture is



- Ports are located on line cards
- FIB is produced by Routing processor and downloaded to the line cards
- Switching fabric is a bus or shared memory

S38.121/RKa s-01

5-69

FIB Lookup speed is determined by nrof reads

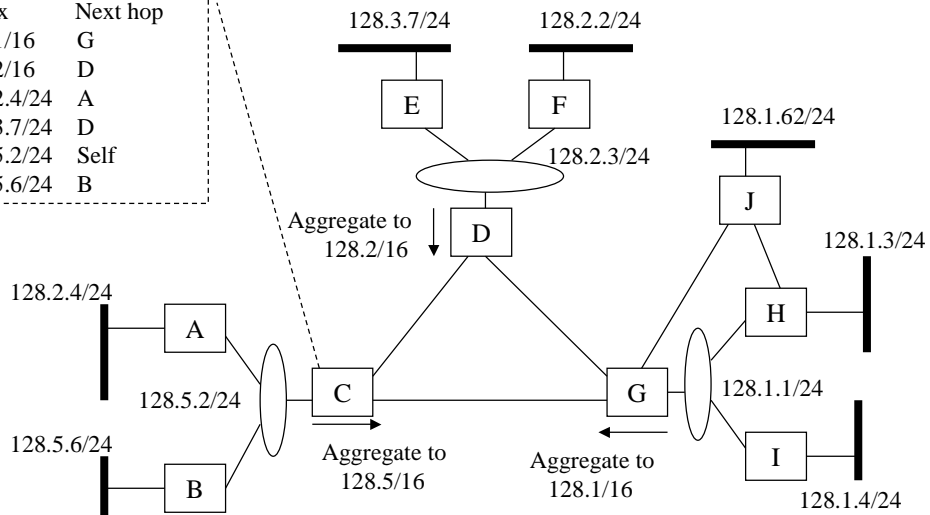
- E.g. access time $30\text{ns} \times 8 \text{ reads} = 240 \text{ ns} \implies >4 \text{ M lookups/s}$
- May need to backtrack --> poor worst case performance.
- Ways to improve performance:
 - Hardware oriented techniques
 - Table compaction techniques
 - e.g. long trie branches with few leaves are packed into a node
 - Hashing techniques
 - problem is unknown mask length -->
 - (e.g. binary search on prefix length)

S38.121/RKa s-01

5-70

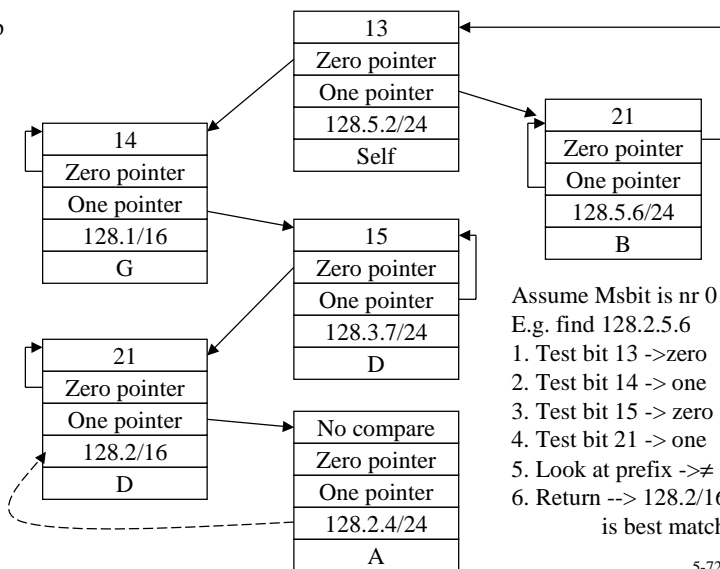
Route aggregation example

Prefix	Next hop
128.1/16	G
128.2/16	D
128.2.4/24	A
128.3.7/24	D
128.5.2/24	Self
128.5.6/24	B

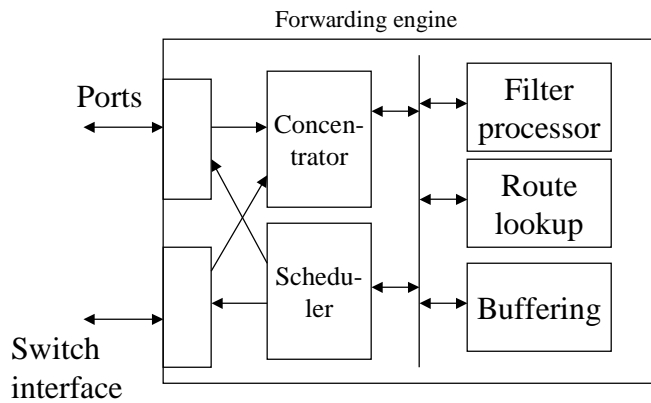


Route lookup may be based on Patricia tree

Prefix	Next hop
128.1/16	G
128.2/16	D
128.2.4/24	A
128.3.7/24	D
128.5.2/24	Self
128.5.6/24	B



Forwarding speed can be increased by parallel processing



Example based on Bell Labs prototype.

Boxes are 33MHz...
66MHz FPGAs.

Can process all headers
prior to buffering at
1Gbit/s line speeds!

--> can provide QoS.

$$\text{Forw time} = 40 \times 8 \text{ b} / 1\text{G/s} = 320 \text{ ns.}$$